

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representation of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY

**As rescanning documents *will not* correct
images, please do not report the images to the
Image Problem Mailbox.**

**JAPANESE PATENT APPLICATION,
FIRST PUBLICATION No. H4-102152**

INT. CL.⁵: G06F 13/00
H04L 29/08

PUBLICATION DATE: April 3, 1992

TITLE Virtual Channel Communication System
APPLICATION NO. H2-219364
FILING DATE August 21, 1990
APPLICANT(S) MITSUBISHI ELECTRIC CORP.
INVENTOR(S) Toshiji AIURA and Yuji NIMURA

CLAIMS

1. A virtual channel communication system wherein a virtual channel is constructed between a first and second computer between which mutual communications are performed, the virtual channel communication system being characterized by comprising: a first step of a first computer requesting a second computer which is its communication partner to put communications on hold; a second step of the second computer which has consented to putting communications on hold putting a condition of connection with the first computer over a virtual channel on hold; a third step of the first computer undoing the aforementioned hold on communications by the second computer and requesting resumption of communications; and a fourth step of the second computer undoing the aforementioned hold on communications with the first computer, resuming a condition of connection of the virtual channel and resuming communications.

2. A virtual channel communication system wherein a virtual channel is constructed between a first and second computer between which mutual communications are performed, the virtual channel communication system being characterized by comprising: a first step of a first computer requesting a second

computer which is its communication partner to put communications on hold; a second step of the second computer which has consented to putting communications on hold putting a condition of connection with the first computer over a virtual channel on hold and temporarily suspending the software in the second computer; a third step of the first computer undoing the aforementioned hold on the connection state with the second computer and requesting resumption of communications; and a fourth step of the second computer undoing the aforementioned connection state with the first computer, undoing the temporary suspension of the software in the second computer, resuming a condition of connection of the virtual channel and resuming communications.

DETAILED DESCRIPTION OF THE INVENTION

Field of Industrial Application

The present invention relates to a virtual channel communication system which achieves a simulated channel connection between two computers.

Prior Art

As an example of a conventional virtual channel communication system of this type, there is the socket mechanism described in UNIX BSD 4.3 Programmer's Manual Reference Guide SOCKET 2 which complies with the RFC 793, Transmission Control Protocol. This shall be explained using the block diagram shown in Fig. 2 and the flow chart shown in Fig. 3.

In Fig. 2, the first computer 1 and the second computer 2 are connected to the communication path 3 and perform data communications with each other.

Upon request by software 6 operating on the first computer 1, the communication portion 4 sets up a virtual channel between the software 6 and the software 7 via the communication portion 5 of the second computer 2 (step S1). Next, the communication portion 4 enters a state of awaiting an event from the communication path 3 and the software 6 (step S2).

Here, if there is received data from the communication path 3 (step S3), then the data are received (S4 of Fig. 3), and if the received data does not constitute a request to end communications (step S5), then the required procedures are performed to transfer information to the software (step S6), an event standby state is reentered (step S2), and when the received data constitutes a request to end communications (step S5), a

severance of the virtual channel is performed (step S10).

On the other hand, if there is data transmitted from the software 6, the data is transmitted (step S7), and if it is successfully transmitted (step S8), an event standby state is reentered (step S2). If the transmission is not successful, it is decided that a malfunction has occurred on the communication path 3 or the second computer 2, and the virtual channel is severed (step S10).

Furthermore, if the time over which event standby is performed exceeds a predetermined time, a connection confirmation signal is transmitted (step S9), and if this is successfully transmitted (step S8), an event standby state is reentered, whereas if the transmission is not successful, it is decided that a malfunction has occurred on the communication path 3 or in the second computer 2, and the virtual channel is severed (step S10).

It is made possible to detect when a malfunction has occurred on the communication path 3 or in the second computer 2 in this way, thus increasing the reliability of virtual channel connections. The communication path 5 of the second computer 2 also operates in a manner similar to the above, thus enabling malfunctions of the communication path 3 or the first computer 1 to be detected.

Problems to be Solved by the Invention

Since conventional virtual channel systems make the above-described operations, when for example, the connection of the first computer 1 is temporarily severed from the communication path 3, it is not possible to confirm the connection between the first computer 1 and the second computer 2, so that the communication portions 4 and 5 cannot maintain the virtual communication channel established between the software 6 and 7, and it must be discontinued. For this reason, when the first computer 1 and the communication path 3 are to be temporarily severed, the operations of the software 6 must be temporarily ended, the first computer 1 and the communication path 3 must be severed, and subsequently, the software 6 must be restarted after reconnecting the communication path 3. As a result, there is a problem in that the software 6 cannot be operated continuously in a state immediately following severance of the first computer 1 and the communication path 3.

In order to resolve this problem, the present invention maintains and holds the virtual channel connection even if the connection of the communication path 3 which links the first computer 1 and the second computer 2 is severed, reproduces the state prior to the hold on the computer after reconnection to the communication path 3, thereby enabling the operations of the software 6 and 7 to be performed continuously.

Means for Solving the Problems

In order to resolve these problems, the present invention comprises a first step of a first

computer 1 requesting a second computer which is its communication partner to put communications on hold (step P11); a second step of the second computer 2 which has consented to putting communications on hold putting a condition of connection with the first computer over a virtual channel on hold (step P5); a third step of the first computer 1 undoing the aforementioned hold on communications by the second computer 2 and requesting resumption of communications (step P21); and a fourth step of the second computer 2 undoing the aforementioned hold on communications with the first computer 1, resuming a condition of connection of the virtual channel and resuming communications (step P16).

Additionally, another invention of the present application is such that in the second step (step P5) of the above-described invention, the software 7 of the second computer 2 is temporarily suspended (step P18), and in the fourth step (step P16), the temporary suspension of the software 7 of the second computer 2 is undone (step P17), the connection state of the virtual channel is reproduced, and the communication is resumed.

Functions

In the virtual channel communication system of this invention, according to a first step (step P11), the first computer 1 requests of the second computer 2 which is its communication partner to put communications on hold, according to the second step (step P5), the second computer 2 which has consented to the hold on communications puts the condition of connection with the first computer 1 on hold, according to the third step (step P21), the first computer 1 releases the hold on communications on the second computer 2 and requests resumption of communications, and according to the fourth step (step P16), the second computer 2 releases the hold on the state of communication of the first computer 1, reproduces the connection conditions of the virtual channel, and resumes communications.

In the virtual channel communication system of a different invention of the present application, the second computer 2, after going through the first step (step P11) and consenting to holding communications in the second step (step P5), holds the state of connection of the virtual channel with the first computer 1 and temporarily suspends the software 7 in the second computer (step P18), then after passing through the third step (step P21), releases the first computer 1 from the above-described state of connection in the fourth step (step P16), and undoes the temporary suspension of the software 7 in the second computer 2 (step P17), then reproduces the connection condition of the virtual channel and resumes communications.

Embodiments of the Invention

Herebelow, an embodiment of the present invention shall be explained in detail using Figs. 1(a), (b) and (c) and Fig. 2. Figs. 1(a), (b) and (c) are flow charts showing the operations of the communication portion 4 and communication portion 5 of Fig. 2 in

accordance with the present invention, used for the explanation of the operations of both communication portions 4 and 5. Fig. 3 which is used to explain the prior art is also used to explain an embodiment of the present invention.

The first computer 1 and the second computer 2 connect to the communication path 3 and communicate with each other.

First, upon a request from the software 6, the communication portion 4 establishes a virtual channel between the software 6 and the software 7 via the communication portion 5 of the second computer 2 (step P1). Next, the communication portion 4 enters a standby state of awaiting an event from the communication path 3 and the software 6 (step P2). The above-described operations are performed similarly in the communication portion 5.

Next, the operations of the communication portion 4 for the case where normal data communications are performed shall be explained. If there is received data from the communication portion 5 at the communication portion 4 (step P3), the data are received (step P4), and if the data are normal data which are not a virtual channel hold request (step P5) nor indicate termination of communications (step P6), they are transferred to the software 6 (step P7) and an event standby state is reentered (step P2). If the received data indicates the end of data (step P6), the virtual channel is severed (step P12).

Then, if the communication portion 4 has data to transmit to the communication portion 5 (step P3), the data are transmitted (step P8), and if the data are transmitted successfully (step P9) and the transmitted data do not constitute a virtual channel hold request (step P11), then an event standby state is reentered (step P2). If the transmitted data are not successfully transmitted (step P9), it is determined that a malfunction has occurred on the communication path 3 or the second computer 2, and the virtual channel is severed (step P12).

Furthermore, if the time of the event standby state of the communication portion 4 exceeds a predetermined period (step P10), then a connection confirmation signal is sent (step P3), and if this is able to be transmitted successfully (step P9), the transmitted data do not constitute a virtual channel hold request signal (step P11), so an event standby state is reentered (step P2), whereas if it is not able to be transmitted successfully (step P9), then it is determined that a malfunction has occurred on the communication path 3 or the second computer 2, and the virtual channel is severed (step P12). The above operations are the same in the case of a normal communication state of the communication portion 5.

Here, the operations of the communication portion 4 from when the communication portion 4 requests of the communication portion 5 a hold of the virtual channel in the normal communication state until it requests a release of the hold on the virtual channel and returns to the normal communication state shall be explained using Figs. 1(a), (c)

and Fig. 2. The communication portion 4 sends data indicating a virtual channel hold request to the communication portion 5 (step P8), and if the data are transmitted successfully (step P9), the virtual channel goes into a hold state (step P11). Then, the communication portion 4 goes into event standby (step P19), during which time even if data are received from the communication portion 5, they are ignored because the virtual channel is held (step P20). When data to be transmitted from the software 6 exists in the communication portion 4 (step P20), if the data is data indicating a release of the hold of the virtual channel (step P21), and as a result of transmission (step P22) is successfully transmitted (step 23), then the hold of the virtual channel is released, and an event standby state for normal communications is entered (step P2). Even if an attempt is made to transmit data other than a virtual channel release request from the communication portion 4 during the virtual channel hold, it is not transmitted (step P21) and an event standby state is entered (step P19). Additionally, also when the virtual channel hold release request is not transmitted successfully (step 23), an event standby state is entered (step P19). The operations of the communication portion 5 from when the communication portion 5 requests a hold of the virtual channel with respect to the communication portion 4 until the hold is released and the normal communication state is resumed is the same as above.

The operations of the communication portion 5 corresponding to the operations of the communication portion 4 from when the communication portion 5 receives a virtual channel hold request in a state of normal communications from the communication portion 4 until the reception of a virtual channel hold release request and resumption of the normal communication state shall be explained using Figs. 1(a) and (b) and Fig. 2. If the data received from the communication portion 4 by the communication portion 5 (step P4) is data indicating a virtual channel hold request (step P5), then the communication portion 5 puts the virtual channel on hold and goes into event standby (step P13). When there is data transmitted from the software 7 in the communication portion 5 (step P14), the software 7 is temporarily suspended (step P18) and an event standby state is entered (step P13). If the communication portion 5 has received data from the communication portion 4 (step P14), the data are received (step P15), and if the received data are data indicating a virtual channel hold release (step P16), the temporary suspension of the software is undone (step P17) and an event standby state for normal communications is entered (step P2). If data other than a virtual channel hold release request is received during the virtual channel hold (step P16), an event standby state is entered without performing anything, in the virtual channel hold state (step P13). The operations of the communication portion 4 from when the communication portion 4 receives a virtual channel hold request from the communication portion 5 in a normal communication state until a virtual channel hold release request is received and a normal communication state is resumed are the same as above.

By performing the hold of the virtual channel in the above-related manner, even if the connection between the computer and the communication path is severed, a severance of the virtual channel as in the conventional system does not occur, so that, for example,

in a portable computer such as a laptop computer, it is possible to realize a resume function such that even if the communication path is severed and the computer is carried away to a different location, the work done immediately prior to severance can be resumed by reconnecting the transmission path, whereby the time required for startup and shutdown of the computer system as well as termination, initiation and establishment of software for performing communications can be shortened.

It is possible to append an encryption key when transmitting the virtual channel hold request (step P11) and to append an encryption key when requesting virtual channel hold release, making it impossible to release the hold unless the two match, thereby enabling the addition of a security function. Furthermore, during the process of temporarily suspending the operation of the software (step P18), it is possible to return to the software an error indicating that the virtual channel is on hold.

Effects of the Invention

As described above, according to the present invention, according to a first step, the first computer requests of the second computer which is its communication partner to put communications on hold, according to the second step, the second computer which has consented to the hold on communications puts the condition of virtual connection with the first computer on hold, according to the third step, the first computer releases the hold on communications on the second computer and requests resumption of communications, and according to the fourth step, the second computer releases the hold on the state of communication of the first computer, reproduces the connection conditions of the virtual channel, and resumes communications, whereby it is possible to maintain and hold the virtual channel even if the connection of the communication path connecting the first computer and the second computer is severed, and to reproduce the state prior to the hold of the computer after reconnection to the communication path. Additionally, according to another invention, when the connection condition of the second computer is put on hold, the software is suspended, and the software is activated when the hold is released, thus having the effect that work on the software can be resumed after interruption.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1(a), (b) and (c) are operational flow charts of the present invention, Fig. 2 is a structural diagram of a computer communication network for explaining the system of the present invention, and Fig. 3 is an operational flow chart of a conventional example.

1, 2 . . . computer; 3 . . . communication path; 4, 5 . . . communication portion; 6, 7 . . . software.

In the drawings, the same numbers denote the same or corresponding parts.

⑪ 公開特許公報 (A)

平4-102152

⑫ Int. Cl. 5

G 06 F 13/00
H 04 L 29/08

識別記号

352

庁内整理番号

7368-5B

⑬ 公開 平成4年(1992)4月3日

8020-4M H 04 L 13/00 307 Z
審査請求 未請求 請求項の数 2 (全6頁)

⑭ 発明の名称 仮想回線通信方式

⑮ 特 願 平2-219364

⑯ 出 願 平2(1990)8月21日

⑰ 発明者 相浦利治 神奈川県鎌倉市大船5丁目1番1号 三菱電機株式会社情報電子研究所内

⑰ 発明者 二村祐地 神奈川県鎌倉市大船5丁目1番1号 三菱電機株式会社情報電子研究所内

⑰ 出願人 三菱電機株式会社 東京都千代田区丸の内2丁目2番3号

⑰ 代理人 弁理士 宮園純一

明細書

1. 発明の名称

仮想回線通信方式

2. 特許請求の範囲

(1) 第1、第2のコンピュータ間に仮想回線を接続し、相互に通信を行う仮想回線通信方式において、第1のコンピュータがその通信相手である第2のコンピュータに対し通信の保留を要求する第1の手順と、通信の保留を受諾した第2のコンピュータが第1のコンピュータとの仮想回線の接続状況を保留する第2の手順と、第1のコンピュータが第2のコンピュータの上記通信の保留を解除して通信の再開を要求する第3の手順と、第2のコンピュータが第1のコンピュータの上記接続状態の保留を解除して仮想回線の接続状況を再現し、通信を再開する第4の手順とを有することを特徴とする仮想回線通信方式。

(2) 第1、第2のコンピュータ間に仮想回線を接続し、相互に通信を行う仮想回線通信方式において、第1のコンピュータがその通信相手である第

2のコンピュータに対し通信の保留を要求する第1の手順と、通信の保留を受諾した第2のコンピュータが第1のコンピュータとの仮想回線の接続状況を保留し、かつ第2のコンピュータ内のソフトウェアを一時的に停止させる第3の手順と、第1のコンピュータが第2のコンピュータの上記接続状態の保留を解除して通信の再開を要求する第4の手順と、第2のコンピュータが第1のコンピュータの上記接続状態を解除し、かつ上記第2のコンピュータ内のソフトウェアの一時的停止を解除して仮想回線の接続状況を再現し、通信を再開する第5の手順とを有する仮想回線通信方式。

3. 発明の詳細な説明

〔産業上の利用分野〕

この発明は、2つのコンピュータ間に疑似的な回線接続を実現する仮想回線通信方式に関するものである。

〔従来の技術〕

従来のこの種の仮想回線通信方式の例としては RFC 793, Transmission Control Protocol

に掲載したUNIX BSD4.3 プログラマーズ・マニュアル・リファレンスガイド SOCKET 2 に示されるsocket機構がある。これを第2図に示すブロック図及び第3図に示すフローチャートを用いて説明する。

第2図において、第1のコンピュータ1と第2のコンピュータ2は、通信路3に接続され互いにデータ通信を行う。

通信部4は第1のコンピュータ1上で動作するソフトウェア6の要求により第2のコンピュータ2の通信部5を経由して、ソフトウェア6とソフトウェア7の間に仮想回線を設定する(ステップS1)。次に通信部4は通信路3及びソフトウェア6からのイベント待ち状態になる(ステップS2)。

ここで、通信路3からの受信データがあれば(ステップS3)そのデータを受信し(第3図S4)、受信したデータが通信の終了要求でないならば(ステップS5)必要な処理を行ってソフトウェア6に情報を渡し(ステップS6)再びイベ

ント待ちとなり(ステップS2)、受信したデータが通信の終了の要求の場合(ステップS5)、仮想回線の切断を行う(ステップS10)。

一方、ソフトウェア6から送信するデータがあれば、データを送信し(ステップS7)、正常に送信できれば(ステップS8)再びイベント待ち(ステップS2)に戻る。もし、正常に送信できなければ通信路3あるいは、第2のコンピュータ2に何らかの異常が発生したと判断し仮想回線の切断を行う(ステップS10)。

さらにイベント待ちを行っている時間が、ある一定時間を越えたならば接続確認信号を送信し(ステップS9)、正常に送信できれば(ステップS8)再びイベント待ちとなり、正常に送信できなければ通信路3あるいは第2のコンピュータ2に何らかの障害が発生したと判断して仮想回線を切断する(ステップS10)。

このように通信路3あるいは第2のコンピュータ2に障害が発生した場合、感知することができるようにして仮想回線接続の信頼性を向上してい

る。なお、第2のコンピュータ2の通信部5でも上記と同様に動作し、通信路3あるいは第1のコンピュータ1の障害を検知できる。

(発明が解決しようとする課題)

従来の仮想回線方式では以上のような動作となっているので、例えば一時的に通信路3から第1のコンピュータ1の接続を切り離す場合、第1のコンピュータ1と第2のコンピュータ2間の接続確認ができなくなるため、通信部4及び5がソフトウェア6及び7間に確立している仮想通信回線接続を維持できず切断することとなる。このため第1のコンピュータ1と通信路3を一時的に切り離す場合は、ソフトウェア6の動作をいったん終了させ、第1のコンピュータ1と通信路3を切り離し、その後、通信路3を再び接続してから、再度ソフトウェア6の起動をし直さなければならない。このため、ソフトウェア6は、第1のコンピュータ1と通信路3を切断した直後の状態から接続して動作させることができないという問題点があった。

この発明はかかる問題を解消するために、第1のコンピュータ1及び第2のコンピュータ2間をつなぐ通信路3の接続を切断しても仮想回線接続を維持保留し、再び通信路3と接続後コンピュータ上で保留前の状態を再現させ、ソフトウェア6及び7の作業を継続して行えることを可能とするものである。

(課題を解決するための手段)

本発明はこれらの問題点を解決するために、第1のコンピュータ1がその通信相手である第2のコンピュータ2に対し通信の保留を要求する第1の手順(ステップP11)と、通信の保留を受諾した第2のコンピュータ2が第1のコンピュータ1との仮想回線の接続状況を保留する第2の手順(ステップP5)と、第1のコンピュータ1が第2のコンピュータ2の上記通信の保留を解除して通信の再開を要求する第3の手順(ステップP21)と、第2のコンピュータ2が第1のコンピュータ1の上記接続状態の保留を解除して仮想回線の接続状況を再現し、通信を再開する第4の手

順（ステップP16）とを有するものである。

また本願別発明は、上記発明における第2の手順（ステップP5）において第2のコンピュータ2内のソフトウェア7を一時的に停止させ（ステップP18）、第4の手順（ステップP16）において上記第2のコンピュータ2内のソフトウェア7の一時的停止を解除（ステップP17）して仮想回線の接続状況を再現し、通信を再開するものである。

（作用）

この発明における仮想回線通信方式は、第1の手順（ステップP11）により第1のコンピュータ1がその通信相手である第2のコンピュータ2に対し通信の假想を要求し、第2の手順（ステップP5）により通信の保留を受諾した第2のコンピュータ2が第1のコンピュータ1との仮想回線の接続状況を保留し、第3の手順（ステップP21）により第1のコンピュータ1が第2のコンピュータ2の上記通信の保留を解除して通信の再開を要求し、第4の手順（ステップP16）によ

り第2のコンピュータ2が第1のコンピュータ1の上記接続状態の保留を解除して仮想回線の接続状況を再現し、通信を再開する。

本願別発明における仮想回線通信方式は、上記第1の手順（ステップP11）を経て、第2の手順（ステップP5）により通信の保留を受諾した第2のコンピュータ2が第1のコンピュータ1との仮想回線の接続状況を保留し、かつ第2コンピュータ2内のソフトウェア7を一時的に停止（ステップP18）させ、上記第3の手順（ステップP21）を経て、第4の手順（ステップP16）により第1のコンピュータ1の上記接続状態を解除し、かつ上記第2コンピュータ2内のソフトウェア7の一時的停止を解除（ステップP17）して仮想回線の接続状況を再現し、通信を再開する。

（発明の実施例）

以下、本発明の一実施例を第1図(a)、(b)、(c)、及び第2図を用いて詳細に説明する。第1図(a)、(b)、(c)は第2図における通信部4及び通信部5の本発明に係わる動作を表すフローチャートであり、

通信部4及び5の動作説明に共通して用いられる。前記従来技術を説明するために用いた第3図は、本発明の実施例を説明するためにも用いる。

第1のコンピュータ1及び第2のコンピュータ2は、通信路3に接続し、互いに通信を行う。

まず、通信部4はソフトウェア6の要求により第2のコンピュータ2の通信部5を経由して、ソフトウェア6とソフトウェア7の間に仮想回線を設定する（ステップP1）。次に通信部4は通信路3及びソフトウェア6からのイベント待ち状態になる（ステップP2）。以上の動作は通信部5においても同様に行う。

次に、通常のデータ通信を行っている場合の、通信部4の動作を説明する。通信部4に通信部5からの受信データがあれば（ステップP3）、そのデータを受信し（ステップP4）、そのデータが仮想回線保留要求でなく（ステップP5）通信の終了を示すものでない通常データならば（ステップP6）、ソフトウェア6に渡し（ステップP7）、再びイベント待ちとなる（ステップP2）。

受信したデータが通信の終了を示すものならば（ステップP6）、仮想回線を切断する（ステップP12）。

そして、通信部4が通信部5に送信するデータがあれば（ステップP3）、そのデータを送信し（ステップP8）、正常にデータが送信され（ステップP9）送信したデータが仮想回線保留要求でないならば（ステップP11）、再びイベント待ちとなる（ステップP2）。送信したデータが正常に送信されない場合は（ステップP9）通信路3あるいは第2のコンピュータ2に何らかの障害が発生したと判断して仮想回線を切断する（ステップP12）。

さらに、通信部4のイベント待ちを行っている時間が一定時間を越えたならば（ステップP10）、接続確認信号を送信し（ステップP3）正常に送信できれば（ステップP9）送信データは仮想回線保留要求信号ではないので（ステップP11）再びイベント待ちとなり（ステップP2）、正常に送信できなければ（ステップP9）、通信路3

あるいは第2のコンピュータ2に何らかの障害が発生したと判断して仮想回線を切断する(ステップP12)。以上の動作は通信部5の通常通信状態においても同様である。

ここで、通信部4が、通信部5に対し通常の通信状態で仮想回線の保留を要求してから、仮想回線の保留解除を要求し通常の通信状態に戻るまでの、通信部4の動作を第1図(a), (b), 第2図を用いて説明する。通信部4が通信部5に対し仮想回線保留要求を示すデータを送信し(ステップP8)データが正常に送信されたならば(ステップP9)仮想回線は保留の状態となる(ステップP11)。そして、通信部6はイベント待ちとなり(ステップP19)、この間、通信部5からのデータを受信しても(ステップP20)仮想回線が保留中なので無視される。通信部4にソフトウェア6からの送信するデータが存在する場合(ステップP20)、そのデータが仮想回線の保留解除を示すデータであり(ステップP21)、送信した結果(ステップP22)正常に送信されたならば(ス

テップP23)、仮想回線の保留は解除され通常の通信におけるイベント待ち状態になる(ステップP2)。仮想回線保留中に、仮想回線解除要求以外のデータを通信部4から送信しようとしても、送信されずに(ステップP21)イベント待ちとなる(ステップP19)。また、仮想回線保留解除要求が正常に送信されなかった場合にも(ステップP23)、イベント待ちとなる(ステップP19)。通信部5が通信部4に対し仮想回線の保留を要求してから、保留の解除を行い通常の通信状態に戻るまでの通信部5の動作も以上と同様である。

通信部4の動作に対応して通信部5が通信部4より通常の通信状態で仮想回線の保留要求を受けた後、仮想回線保留解除の要求を受け、通常の通信状態に戻るまでの通信部5の動作と同じく第1図(a), (b)及び第2図を用いて説明する。通信部5が通信部4から受信したデータが(ステップP4)仮想回線保留要求を示すデータであったならば(ステップP5)、通信部5は、仮想回線を保

留状態としイベント待ちになる(ステップP13)。通信部5にソフトウェア7からの送信するデータがあった場合には(ステップP14)、ソフトウェア7を一時的に停止させ(ステップP18)イベント待ちとなる(ステップP13)。通信部4からのデータを通信部5が受信したならば(ステップP14)そのデータを受信し(ステップP15)、受信データが仮想回線の保留解除を示すデータならば(ステップP16)、ソフトウェアの一時停止を解除し(ステップP17)通常の通信におけるイベント待ち状態となる(ステップP2)。仮想回線保留中に仮想回線保留解除要求以外のデータを受信した場合(ステップP16)、何も行わずに再び仮想回線保留状態のままイベント待ちとなる(ステップP13)。通信部4が通信部5より通常の通信状態から仮想回線の保留を要求を受けてから、仮想回線保留解除の要求を受け、通常の通信状態に戻るまでの通信部4の動作も以上と同様である。

以上のように仮想回線の保留を行うことにより、

コンピュータと通信路の接続を切断したとしても、従来方式のような仮想回線の切断が起らなくなり、例えば、ラップトップコンピュータのような持ち運び可能なコンピュータで、通信路を切断し持ち込んだ後、作業場所を変えて再度通信路に接続すれば切断直前の作業を引き続きすぐに行えるレジューム機能の実現が可能となり、コンピュータシステムの立ち上げ、立ち下げ、通信を行うソフトウェアの終了、起動、設定等の時間が短縮される。

なお、仮想回線保留要求の送信時に(ステップP11)暗号キーを付加し、仮想回線保留解除要求時にも暗号キーを付加し、両者が一致しないと保留解除できないようにすることで、セキュリティ機能を付加してもよい。さらにソフトウェアの動作を一時的に停止させる処理において(ステップP18)、仮想回線が保留中であることを示すエラーをソフトウェアへ返してもよい。

(発明の効果)

以上のように、この発明によれば、第1の手順

により第1のコンピュータがその通信相手である第2のコンピュータに対し通信の保留を要求し、第2の手順により通信の保留を受諾した第2のコンピュータが第1のコンピュータとの仮想回線の接続状況を保留し、第3の手順により第1のコンピュータが第2のコンピュータの上記通信の保留を解除して通信の再開を要求し、第4の手順により第2のコンピュータが第1のコンピュータの上記接続状態の保留を解除して仮想回線の接続状況を再現し、通信を再開するようにしたので、第1のコンピュータ及び第2のコンピュータ間をつなぐ通信路の接続を切断しても仮想回線を維持保留し、再び通信路と接続したコンピュータ上で保留前の状態を再現させることができる。また別発明によれば第2のコンピュータが接続状況を保留するときにそのソフトウェアを停止させ、この保留を解除するときにソフトウェアを駆動するようにしたのでソフトウェアでの作業を、中断後連続して行える効果がある。

4. 図面の簡単な説明

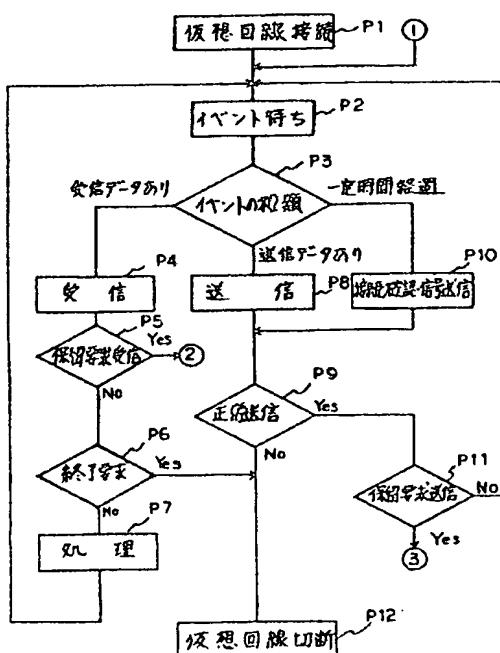
第1図(a), (b), (c)は本発明の動作フローチャート、第2図は本発明方式を説明するためのコンピュータ通信網の構成図、第3図は従来例の動作フローチャートである。

1, 2 … コンピュータ、3 … 通信路、
4, 5 … 通信部、6, 7 … ソフトウェア。

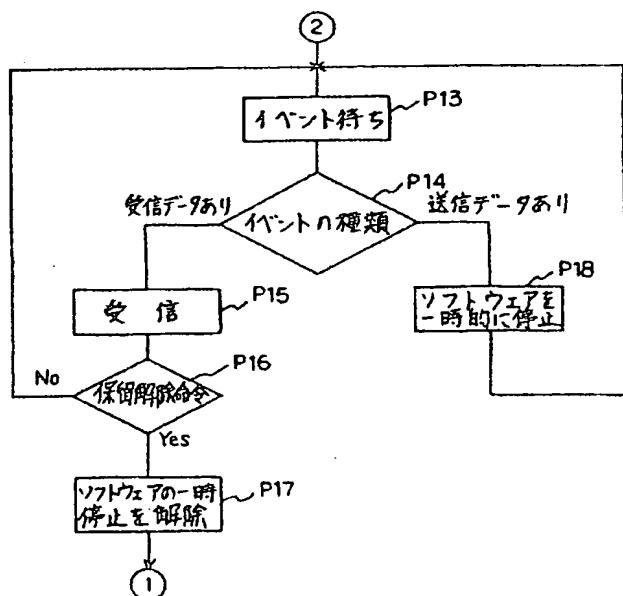
なお図中、同一番号は、同一または相当する部分を示す。

代理人弁理士 宮園純一

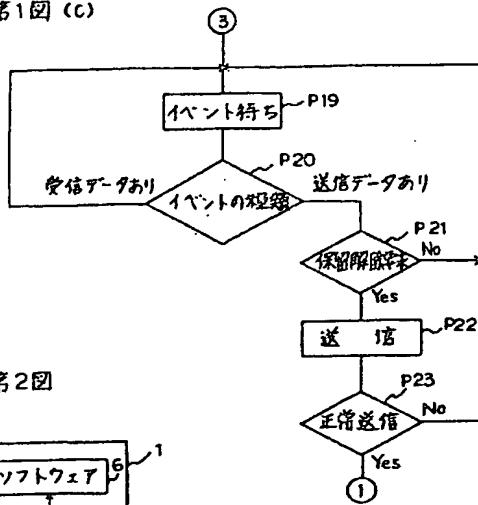
第1図 (a)



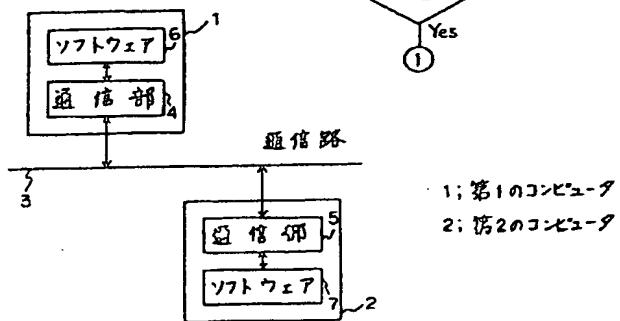
第1図 (b)



第1図 (C)



第2図



第3図

